A Robust and Secure Methodology for Network Communications

Mayank Srivastava[1], Mohd. Qasim Rafiq [2] and Rajesh Kumar Tiwari[3]

[1] Department of Computer Science, GLAITM
Mathura, U.P. 281406, India

[2] Department of Computer Engg, Zakir Husain College of Engg. & Technology, AMU
Aligarh, U.P., India

[3] Department of Computer Science, GLAITM
Mathura, U.P. 281406, India

Abstract

With the rapid development of a network multimedia environment, digital data can now be distributed much faster and easier. To maintain privacy and security cryptographic alone is not enough. In recent years, steganography has become attractive vicinity for network communications. In this paper, an attempt is made to develop a methodology which calculate the variance of secret message (which the sender wishes to send) and accordingly create a carrier file. This carrier file can be sent in any network (secure & unsecure) without giving any doubt in the attackers mind. The practical implementation of this has been done on Microsoft platform. Experimental results show the feasibility of the proposed techniques.
.

Keywords: Steganography, Cryptography, Image file, Stego file**.**

1. Introduction

The growing use of Internet need to store, send and receive personal information in a secured manner. For this, we may adopt an approach that can transfer the data into different forms so that their resultant data can be understood if it can be returned back into its original form. This technique is known as encryption. However, a major disadvantage of this method is that the existence of data is not hidden. If someone gives enough time then the unreadable encrypted data may be converted into its original form.
A solution to this problem has already been achieved by using a technique named with the Greek word “steganography” giving a meaning to it as ‘writing in hiding’. The main purpose of steganography is to hide data in a cover media so that other will not notice it [10]. Steganography has been used in various forms for last 2500 years. It has found use in variously in military, diplomatic, personal and intellectual property applications. There are examples from history that serve to illustrate the desire to hide messages or some type of intelligence from others. Histaiacus shaved the head of a messenger, wrote a note encouraging Aristagoras of Miletus to revolt against the king of Persia. After the messenger’s hair grew back, the messenger was dispatched with the message. Obviously, this message wasn’t especially time constrained. Another human vector example includes writing messages on silk, which would then be compressed into a ball and covered with wax. The messenger would swallow the wax ball. The characteristics of the cover media depends on the amount of secret data that can be hidden, the perceptibility of the cover media and its robustness [4,5,6,7,10,18].

Publishing and broadcasting fields also require an alternative solution for hiding information. Unauthorized copying is hot issue in the area like music, film, book and software. To overcome this problem some invisible information can be embedded in the digital media in such a way that no one can easily extract it [1,2,4]. Analogously, software industries have taken advantage of another form of steganography, called watermarking, which is used to establish ownership, identification, and provenance [3,7].

2. Related Works

The most suitable cover media for Steganography is image on which numerous methods have been designed. The main reason is the large redundant space and the possibility of hiding information in the image without attracting attention to human visual system. In this respect, a number of techniques have been developed [1,7] using features like

- Substitution
- Masking and Filtering
- Transform Technique

The method of substitution generally does not increase the size of the file. Depending on the size of the hidden image, it can eventually cause a noticeable change from the unmodified version of the image [4,6]. Least Significant Bit (LSB) insertion technique is an approach for embedding information in a cover image. In this case, every least significant bit of some or all of the bytes inside an image is changed to a bit of the sink image. When using a 24-bit image, one bit of each of the primary color components can be used for the above purpose.

The masking and filtering techniques starts with the analysis of the image. Next, we find the significant areas, where the hidden message will be more integrated to cover the image and lastly we embed the data in that particular area.

In addition to the above two techniques for message hiding, transform techniques has also been employed in embedding the message by modulating coefficients in a transform domain. As an example, we may mention here that Discrete Cosine Transform works by using quantization on the least important parts of the image in respect to the human visual capabilities.

The poor quality of recover image is the main drawback with the proposed LSB algorithm by Anderson and Petitcolas [11]. Raja et.al. [12] have proposed the least significant bit-embedding algorithm where cover image should always at least eight times larger than the sink image. As a result of which it can be pointed out here that a small image may not keep a large image. Marvel [6] has proposed a spread spectrum image steganographic technique for image carrier and able to conceal 5 kB of secret data in 512 x 512 image size. Julio C. Hernandez-Castro [7] has given a concept for steganography in games where the less robust technique saves only few parts of total carrier files. Raja [8] concluded that three secret bits can be embedded in one pixel. Amin [10] has given a secured information hiding system which can only embed 60 kB message in 800 x 600 pixels image.

In conclusion, we can say that all of the above methods used for embedding secret data works on the principle of replacement of entire or some parts of the carrier file. Due to this replacement policy, however, we are not able to conceal high amount of secret data into the carrier file. Further, the existence of carrier file gives a chance for comparing and ultimately one tries to recover the hidden data from the corresponding stego carrier file being created. To overcome these problems, we have designed a robust and secure methodology for network communications. Here, based on the secret data we create our own image file. Robustness, security and high capacity are the major advantages with this methodology. For practical implementation of the various steps of the proposed methodology, we have framed an image creator (IC) and secret data observer (SDO). The paper is organized as follows. The proposed method is discussed in Section 3. Section 4 describes the process of data retrieval. Practical implementation and discussion about the proposed method is given in Section 5 followed by conclusion in Section 6.
provenance [3,7].

3. Proposed Methodology

It is the principle concept in steganography that one has to take care to conceal the secret data in a carrier file such that the combination of both the carrier and the information embedded will never raise any doubt over it. This carrier file can be sent in any communication media for secured communication.

To achieve this, we propose here a model for communication channel. The model is divided into two parts. The first part (Sender Part) of the model takes the secret data (which sender wish to send) make analysis and accordingly creates an image file and sends to the receiver end. The second part (Recipient Part) of the model receives the image (stego image) and retrieves the secret data and displays. The basic model of stego image file creation is shown in Figure 1.

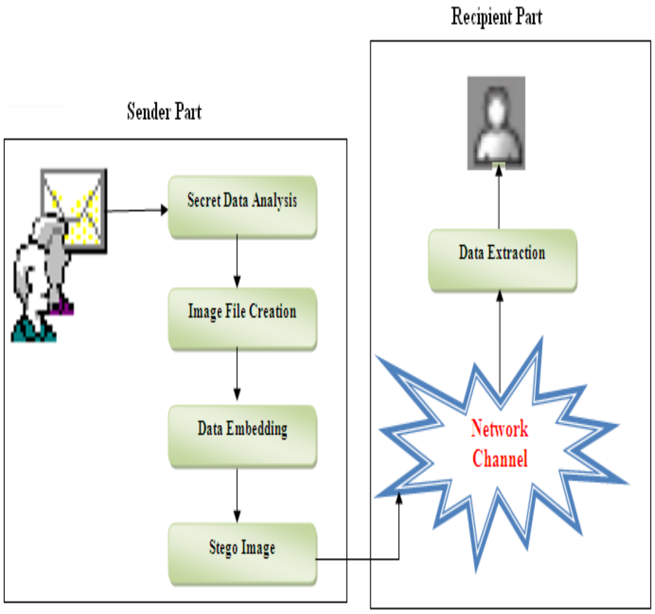
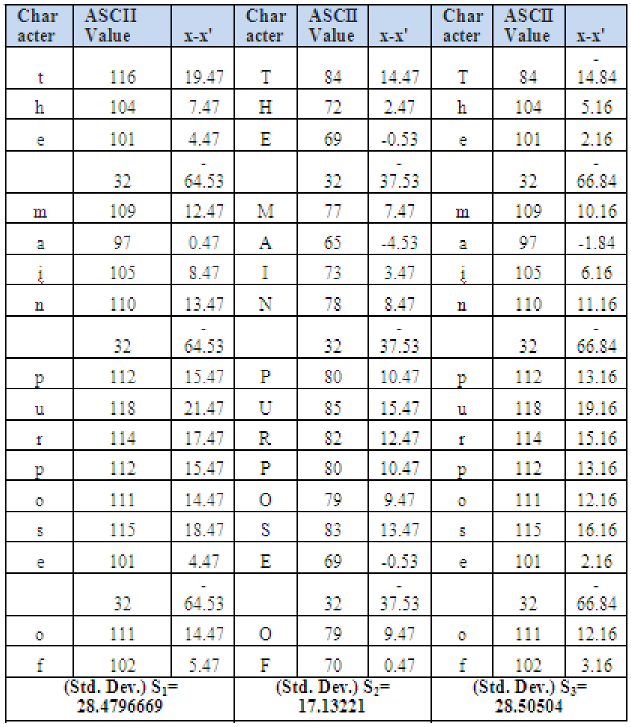


Fig. 1 Basic Model.

3.1 Sender Part

Secret data, which the sender wishes to keep confidential, is the original message or data that is fed into the algorithm as input. Here, we start our process by analyzing the secret data. Initially we calculate standard deviation of the secret data by converting it into three different combinations. In the first arrangement, we take the original secret data (i.e. without changing lower case to upper case or vice versa) and compute the standard deviation. The second combination is made by changing all secret data into lowercase characters. The last combination converts the secret data into corresponding uppercase characters. One can find the standard deviation of an entire population (all secret characters) in cases where every member of a population is sampled. In cases where that cannot be done, the standard deviation S_k is estimated by examining a random sample taken from the population (Fig.2). Example is given below.



Char acter	ASCII Value	x-x'	Char acter	ASCII Value	x-x'	Char acter	ASCII Value	x-x'
t	116	19.47	T	84	14.47	T	84	-14.84
h	104	7.47	H	72	2.47	h	104	5.16
e	101	4.47	E	69	-0.53	e	101	2.16
	32	-64.53		32	-37.53		32	-66.84
m	109	12.47	M	77	7.47	m	109	10.16
a	97	0.47	A	65	-4.53	a	97	-1.84
i	105	8.47	I	73	3.47	i	105	6.16
n	110	13.47	N	78	8.47	n	110	11.16
	32	-64.53		32	-37.53		32	-66.84
p	112	15.47	P	80	10.47	p	112	13.16
u	118	21.47	U	85	15.47	u	118	19.16
r	114	17.47	R	82	12.47	r	114	15.16
p	112	15.47	P	80	10.47	p	112	13.16
o	111	14.47	O	79	9.47	o	111	12.16
s	115	18.47	S	83	13.47	s	115	16.16
e	101	4.47	E	69	-0.53	e	101	2.16
	32	-64.53		32	-37.53		32	-66.84
o	111	14.47	O	79	9.47	o	111	12.16
f	102	5.47	F	70	0.47	f	102	3.16
(Std. Dev.) S_1= 28.4796669			(Std. Dev.) S_2= 17.13221			(Std. Dev.) S_3= 28.50504		

The above examples suggest that the second combination is most suitable since the deviation is comparatively small. Here, we combine three characters of the transmitting data to make a single pixel of the cover shapes. So, the deviation and corresponding ASCII values are the parameters for deciding the pixels of cover image. We then move to the next stage

3.1.1 Image File creation:

Digital images are commonly of two types

1. 8-bit images
2. 24-bit images.

These images can be logically divided into two parts named as header part and body part. The header part consists of all preliminary information like file name, file type, compression type, etc. and the body part keeps the information of pixels. The image editing software, mathematical software and electrical and electronic circuit designing software tools are the best source for generating the header part and body part of the carrier image file. The file header part does not play any primary role in data embedding. We mainly concentrate on the body part of the carrier image file to hide the transmitting data. Thus, some part of the carrier image file can be constructed by geometrical shapes, hand sketch objects, different curves, etc. Generally, methods generating plain curves are useful for creating 2D drawing for general proposes, but for the purpose of engineering application methods for smooth space curves are more suitable and provide more realistic picture (image) of any real-world object. Again, the space curves may be needed in many computer graphics application for representing 3D and 2D objects. Real-world objects around us are mostly 3D and inherently smooth. Our own vision system is best suited to observe 3D objects. The computer display of 3D objects made up of curved surfaces is expected to look natural, reflecting real-world qualities. Therefore, much of computer graphics and manufacturing processes, data plotting, artistic sketch and animation all need spatial smooth curves for achieving accurate representation of real-world objects. By selecting the pre-existing shapes and generating the new objects, we can create the body part of the carrier image file. MatLab, CircuitMaker, Mathematica, Multisim, image editing software, etc. are the software's by which this can be achieved. This solves our purpose by making the body part of the carrier image (see e.g. Fig 3).

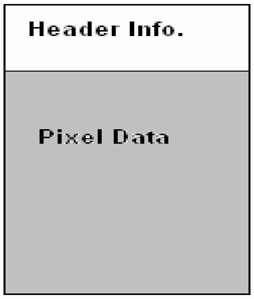


Fig. 3 Image file format

One of the major advantages of this new methodology is independencies of the size of the carrier file, that is, based on the size of the input transmitting data, we generate the required image file. For just an illustration we have shown a normal image in Fig. 4

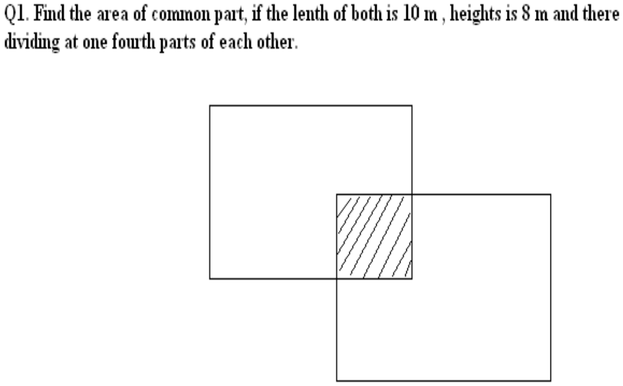

Q1. Find the area of common part, if the lenth of both is 10 m , heights is 8 m and there dividing at one fourth parts of each other.

Fig. 4 A hand sketched image

Further, shading, image rendering, texturing are the tools that can be used to make the image more realistic. Shaded images with a high degree of visual realism create the impression that the images are realistic. Once the image is formed, we start with different image realistic work involving shading the image surfaces and then analyze from the viewer's point of view. The method of texturing also plays a vital role in adding value to a model as shown in Figure 5.

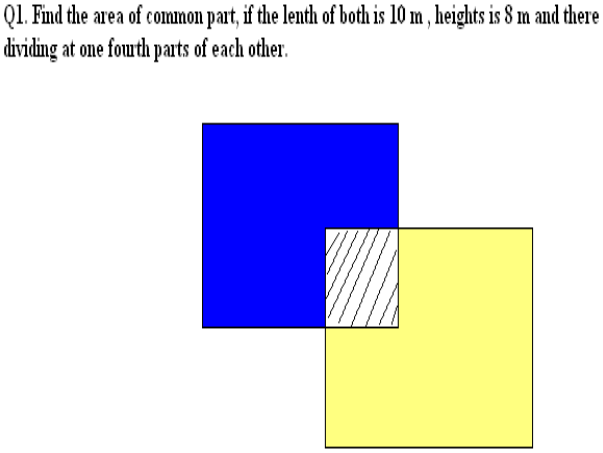

Q1. Find the area of common part, if the lenth of both is 10 m , heights is 8 m and there dividing at one fourth parts of each other.

Fig. 5. Image after certain effects

The stego key is known as password, which is a protection fence for this methodology. After the creation of carrier image file, we start the analysis of the distribution place of the carrier image file and the secret data. Here, we can introduce another layer of protection securities by applying the mathematical formula. The secret data distribution process begins with the given mathematical formula that calculates the next pixel address location for the secret data.

$$M_j = i + (j - 1)*Y, \quad j = 1, 2, 3$$

where M_j is the target location, i is the staring location place, j gives the secret bit number and Y is the distance between two consecutive location in carrier image file. The series of location address for secret character can be defined when $i = 1$ and $Y = 10$, as 1, 11, 21, 31, 41, 51….

For other requirements, we store the value of M_j in the location buffer. This direct replacement of the pixel value to the corresponding transmitting data character gives a freedom to store three characters in one pixel, whereas, the LSB methods require eight pixel to store same three characters. Further, the extra advantage we can get here is the size independency of the pre-existing carrier file. The required pseudo code is given below.

```
Data Embedding
{
1DArray Cover_iamge = Load ("Cover image", &length);
1D Array Stego_Image [ ];
1D Array Secret_data [ ];
1D Array ASCII_data [ ];
Integer L, I, J, K;
INPUT Secret_data;
/* Input the Data Store Position */
INPUT L;
For i= 0 to Length (Secret_data [ ])
    ASCII_data [ I ] = Secret_data [ i];
End For;
/* Secret Data Storing   */
If Length (Secret_data [ ]) > Length (Cover_image [ ]) Then
      Print "Secret Data size is more, Select other method"
Else  Pixel_Place=L;   J=1;
      For I =L  to Length (Secret_data [ ])
            GetColorValue (&K, Cover_image [Pixel_Place]);
            K= Secret_data[J];
            SetColorValue (Stego_image [Pixel_Place], &K );
            J=J+1;         End For ; End if;
SaveImage (Stego_image [])
RenameImage (Stego_image [ ], Cover_Image [ ] );
Remove Image (Stego_image [ ]);
    }
```

3.1.3 Stego Image Coloring

After the secret data embedding process, we start the colour filling procedure. Our algorithm initiates the first step by the checking of the location buffer information and in the next step, it selects the first pixel location and check its presence in the location buffer. If the location buffer values not match with the pixel address, the algorithm changes the pixel intensities with the user supplied value. We continue this process for the entire shape. Further, we can add other shape, objects or text to make the embedded carrier image to more realistic. Here, we have added an extra text which makes the carrier image file into stego image as shown in Fig.6.

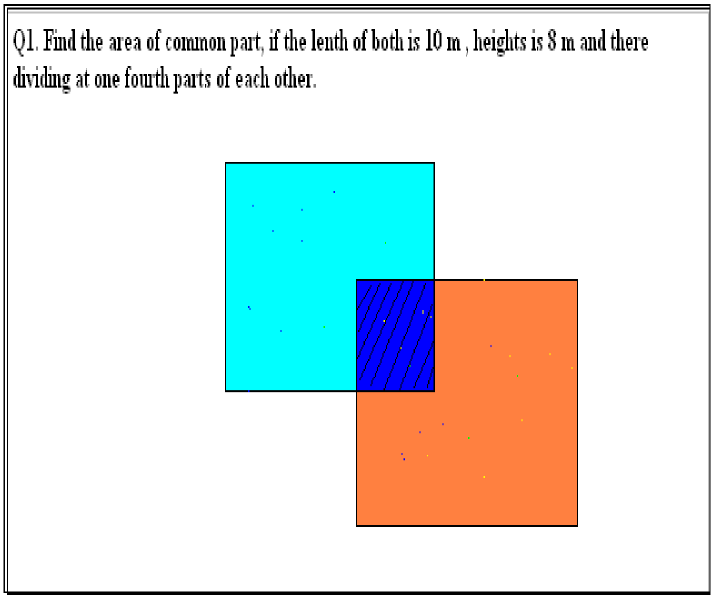

Q1. Find the area of common part, if the lenth of both is 10 m , heights is 8 m and there dividing at one fourth parts of each other.

Fig. 6. Stego Image

3.1.4 Image Transmission

At the end of the process the stego image can be transmitted in any media (i.e. secure and none secure both). Since we are creating a new image for each transaction so any attacker cannot compare the stego file with its original file this gives a good boost for sender and receiver both.

3.2 Data retrieval

The embedded stego image can be retrieved back by using the extraction algorithm. Generally, the extraction is the reverse process of embedding the data into the image.

The received stego image is the primary input of the model. The retrieval process takes the inputs of initial pixel place, stego key and the mathematical formula by which the transmitting data has been distributed. Based on the next pixel location which is obtained by the mathematical formula is generated and the corresponding pixel ASCII value is converted into the characters. Finally, with the aid of a grammar and dictionary, the transmitting data is retrieved and fabricated as shown in Figure 8. Following is the required pseudo code used for the purpose of retrieval of the secret data.

```
Data_Extract ()
{
Array Stego_buffer=Load("stego_image", &Length)
String Secret_txt_buffer;
String Secret_buffer;
Integer I, J, Y, A, L, M;

Integer Red, Blue, Green;

Input I, J, Y;

M=1;

For ( A=1; A <= L; A++ )
   {
Z = i + (j – 1) *Y

IF ( Z== A) The
   {
GetRGB (&Red, &Green, &Blue, Stego_buffer [A]);
Decode (Secret_buffer [M], Red);
Decode (Secret_buffer [M], Green);
Decode (Secret_buffer [M], Blue);
A=A+2;
M=M+1;
  }
     }


For (A=1; A< =M; A++)

{
Secret_txt_buffer [A]= Secret_buffer [M];
}

}
```

4. Practical implementation and discussion

Based on the above methodology, we have designed IC (image creator) and SDO (secret data observer) in Microsoft platform and it may further be implemented in other programming platform like JAVA and C. The initial steps of IC start with the analysis of the secret data; based on the secret data, we select the shapes and objects; and by using some mathematical formula, we distribute it into the carrier image file. The next optional step prompts to user for selecting the colour, pattern and texture filling. Finally, the program asks for any other text, or shape selection for more realistic view of the canvas and at the end stego image is generated. SDO is reciprocal of the IC. We select the stego image and input stego key with the defined mathematical formula. Finally, we get the secret data in output panel.

The three different aspects considered in any information hiding system: capacity, security and robustness. Capacity refers to the amount of information that can be hidden in the cover medium; security refers to an eavesdropper's inability to detect hidden information; whereas, the robustness is referred to the amount of modification that can be made such that the stego medium can withstand before an adversary can destroy the hidden information. In this proposed system, we maintain all above three aspects.

5. Conclusion

Security in secret data transmission has become a very critical aspect of modern internet and intranet systems. In this paper we have designed a novel and secure methodology which creates a new image file, embeds the secret data and transmit the stego file. At the receiver end, the recipient extracts the secret data using the reverse of the embedding process. As this methodology does not depends on the size of the secret data so any one can send any amount of data using any network or media across the world in a completely secured form. As the new methodology maintains two layers of protection, it is hard to detect the secret information by any unauthorized individual.

References

[1]. G. Sahoo and R.K. Tiwari " Designing an Embedded Algorithm for Data Hiding using Steganographic Technique by File Hybridization", IJCSNS International Journal of Computer Science and Network Security, Vol. 8 No. 1 January 2008.

[2]. Donovan Artz " Digital Steganography: Hiding Data within Data", IEEE Internet computing, pp.75-80 May –June 2001.

[3]. Mitchell D.Swanson, Bin Zhu and Ahmed H. Tewfik "TRANSPARENT ROBUST IMAGE WATERMAKING" IEEE 0-7803-3258-x/96. 1996

[4]. M.M Amin, M. Salleh, S.Ibrahim, M.R. Katmin, and M.Z. I. Shamsuddin " Information hiding using Steganography" IEEE 0-7803-7773-March 7, 2003.

[5]. Lisa M. Marvel and Charles T. Retter, "A METHODOLOGY FOR DATA HIDING USING IMAGES", IEEE 0-7803-4506-1/98, 1998

[6]. Bret Dunbar, "A Detailed Look at Steganography techniques and their use in an Open Systems Environment ", January 18, 2002 SANS Institute.

[7]. C. Cachin, "An Information –Theoretic Model for Steganography", inn proceeding 2nd Information Hiding Workshop, vol 1525, pp.303-318, 1998.

[8]. F.A.P Peticolas, R.J. Anderson and M. G. Kuhn, "Information Hiding –A Survey", in proceeding of IEEE, pp. 1062-1078, July 1999.

[9] Venkatraman. S, Ajith Abraham, Marcin Paprzycki " Significance of Steganography on Data Security ", IEEE 0-7695-2108-8, 2004.

[10]. N.F. Johnson and S Jajodia, "Exploring Steganography: Seeing the Unseen ", Computer, vol31, no. 2, Feb 1998, pp. 26-34.

[11] Ross J. Anderson and Fabien A.P. Petitcoals " On the Limits of Steganography", IEEE Journal on Selected Areas in Communications, Vol. 16 NO. 4 MAY 1998.

[12] K.B. Raja, C.R. Chowdary, Venugopal K R, L.M. Patnaik " A Secure Image Steganography using LSB, DCT, and Compression Techniques on Raw Images", IEEE 0-7803-9588-3/05.

[13] S. Craver, On public-key steganography in the presence of an active warden, In Second International Workshop on Information Hiding, Springer- Verlag, 1998.

[14]. A. Allen, Roy (October 2001). "Chapter 12: Microsoft in the 1980's". A History of the Personal Computer: The People and the Technology (1st edition). Allan Publishing. pp. 12–13. ISBN 0-9689108-0-7.
http://www.retrocomputing.net/info/allan/eBook12.pdf.

[15]. Tsung-Yuan Liu and Wen-Hsiang Tsai**, "**A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique", Information Forensics and Security, IEEE Transactions on Volume 2, No. 1, March 2007 p.p 24 – 30.

[16]. Dekun Zou and Shi**,** Y.Q.**, "**Formatted text document data hiding robust to printing, copying and scanning", Circuits and Systems, IEEE International Symposium, Vol.5 , 2005 p.p- 4971 – 4974.

[17]. A Castiglione, A. De Santis, C. Soriente "Taking advantage of a disadvantage: Digital forensics and steganography using document metadata" The Journal of system and software 80 (2007) 750-764.

[18]. Sahoo, G. and Tiwari, R.K. (2010) 'Some new methodologies for secured data coding and transmission', *Int. J. Electronic Security and Digital Forensics*, Vol. 3, No. 2, pp.120–137.

Mayank Srivastava completed his MCA from VBSPU in the year 2002 and currently pursuing M.Tech(Sequential) from GBTU, Lucknow, Uttar Pradesh He is working as a Asst. prof. at GLA Institute of Technology & Mgmt., Mathura, Uttar Pradesh since 2004.

Mohd. Qasim Rafiq completed his M.Tech in the year 1986 from A.M.U, Aligarh, Uttar Pradesh. and Ph.d.in the field of parallel processing in the year 1996 from University of Roorkee, Uttar Pradesh. Currently he is the Chairman of the Department of Computer Engg., ZHCET, AMU, Aligarh, Uttar Pradesh.

Rajesh Kumar Tiwari received his MCA from Nagpur University in 2002 and PhD in the field of Data Security form Birla Institute of Technology, Ranchi in the year 2010. Currently, he is a Reader at GLA Institute of Technology & Mgmt,, Mathura, Uttar Pradesh, India. His research is focused on data security, network security, cloud computing and database management system.